TOC

Language	grammer	(Machine) Automata
1) Regular Language	regular grammer (type 3)	Finite automata (FA)
2. Context free Language	Context free grammer (type 2)	push down automata (PDA)
3. Context Sensitive Language	Context sensitive grammer (type 1)	Linear Bounded Automata (LBA)
4. Recursively enumerable set) (r.e.set)	unrestricted grammer (type 0)	Turing machine (TM)

* Symbol :- is an abstract entity, which can not be defined formally.

1) All lowercase letters
2) All uppercase "
3) All operators
4) All digits
5) All special characters

* string :- It is sequence of symbols written side by side

* length of string :- no. of symbol in a string.

ex $s1 = abc1$

$|s1| = 4$

ex $s2 = b + * @ 123$

$|s2| = 7$

* empty string :- (ϵ) (epsilon)

$$\boxed{|\epsilon| = 0}$$

* prefix of a string :- It is any no. of leading symbols of a string

ex abc1 →

ϵ, a, ab, abc, abc1

ex : ab+@1

ϵ, a, ab, ab+, ab+@, ab+@1

* suffix of a string :- It is any no. of trailing symbols of a string

ex abc1

ϵ, 1, c1, bc1, abc1

* Concatenation of strings :- Joining a two string

→ writing two strings side by side

$$\boxed{s_1 s_2 \neq s_2 s_1}$$

s_1 and s_2 are two string than $s_1 s_2 \neq s_2 s_1$ that is Concatenation of two string is NOT commutative

But

$$\boxed{\epsilon s = s \epsilon = s}$$

* alphabet :- set of symbols.

→ {0, 1} {a, b, c}

{0, 1, a, b}

* formal language :-

It is set of string over an alphabet.

→ all string over alphabet {0,1} which contain exactly ~~1~~ one 0

L1 = {0, 01, 10, 011, 101, 110, 0111, 1011, 1101, 1110, ---}

→ all string over alphabet {a,b} which are ending with a

L1 = {a, ba, aa, aba, aaa, bba, baa, ---}

→ all string over alphabet {0,1} which contain a substring 00

L1 = {00, 100, 001, 000, 0010, 0011, 0001, 0000, 1000, 0100, 1100, 1001

* All the string over the alphabet {0,1} which contains even no. of zero & even no. of ones

{∈, 00, 11, 0011, 1100, 1010, 0101, 1001, 0110, 0000, 1111 ---}

* String over alpabet $\{0,1\}$ start with 0 and end with 1

$L_1 = \{01, 001, 011, 0001, 0011, 0101,$
$0111 \text{ ----- } \}$

→ String over alphabet $\{0,1\}$ with length exactly 4 with which has 1 in 3rd position from left side

$\{\_ \ \_ \ \underline{1} \ \_\}$

$2^3 = 8$

0 0 1 0
0 0 1 1
0 1 1 0
0 1 1 1
1 0 1 0
1 0 1 1
1 1 1 0
1 1 1 1

* <u>Operation on formal language</u> :-

→ relative complement

$L_1 - L_2$ $\boxed{\rightarrow \text{in } L_1 \text{ but not in } L_2}$

absolute complement $\boxed{\overline{L_1} = \Sigma^* - L_1}$

$L_1 \cup L_2$

$L_1 \cap L_2$

first value L1

$\overline{L_1} = \{\epsilon, 1, 00, 110 \ldots\ldots\}$

4) Concatenation of two Language
5) Kleene closure of a Language
6) positive closure of a Language

→ Concatenation of two Language L1 and L2 is L1L2 which contain $\{x y \mid x \in L_1$ and $y \in L_2\}$.

$L1 = \{01, 100, 1011\}$

$L2 = \{10, 1101, 1010\}$

$L_1L2 = \{0110, 011101, 011010, 10010, 1001101, 1001010, 101110, 10111101, 10111010\}$

$$\boxed{L_1L_2 \neq L2L_1}$$

Concatenation of two formal language is not commutative

5) Kleene closure :-

of language L is L^*. which is

$S^0 \cup S^1 \cup S^2 \cup \ldots$ | S^0 ~~...~~ is All possible concatenation or repetition ... order

string with length ~~is~~ '0'

S^1 is all possible string with length '1'.

$L = \{0, 10\}$

$S^0 = \epsilon$

$S^1 = \{0\}$

$S^2 = \{10, 00\}$

$S^3 = \{010, 100, 000\}$

$S^4 = \{0000, 0010, 1000, 1010, 0100\}$

L or any combination

L^*

$L^* = \{\epsilon, 0, 10, 00, 010, 100, 000, 0000, 0010, 1000, 1010, 0100, \ldots\}$

* positive closure of language L is L^+

$L^1 \cup S^2 \cup \ldots$

$$\boxed{L^+ = L^* - \{\epsilon\}}$$

$$\boxed{L^* = L^+ \cup \{\epsilon\}}$$

$L1 = \{0\}$ ① L_1^* ② L_2^*

$L2 = \{11, 1\}$ ③ $L3L_1^*$ ④ $L_1^*L_2$

$L3 = \{10, 011\}$ ⑤ L_2L3

$L_1^* \rightarrow S^0 \cup S^1 \cup S^2$ ---

$S^0 \rightarrow \epsilon$
$S1 \rightarrow 0$
$S2 \rightarrow 00$
$S3 \rightarrow 000$

① $L_1^* \Rightarrow \{\epsilon, 0, 00, 000\}$ ---

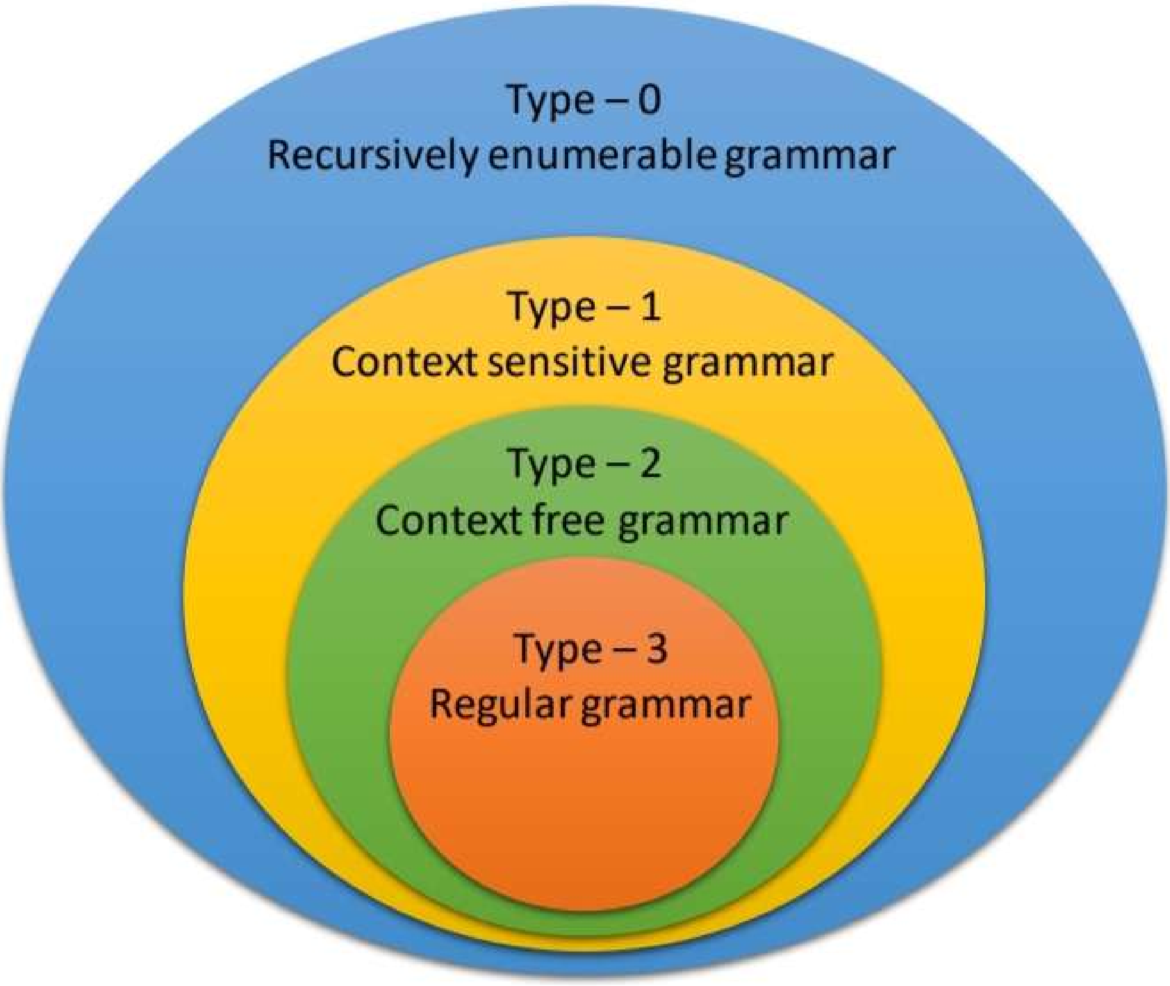
② $L_2^* \rightarrow \{\epsilon, 1, 11, 111, 1111\}$ ---}

③ $L_3 . L_1^* \rightarrow \{$ ~~10, 011, 100, 0110, 1000, 01100~~

$\{10, 100, 1000, 10000$ ---
$011, 0110, 01100, 011000$ ---}

4) $L_1^* L2 \rightarrow \{11, 1, 011, 01, 0011, 001$ ---
}

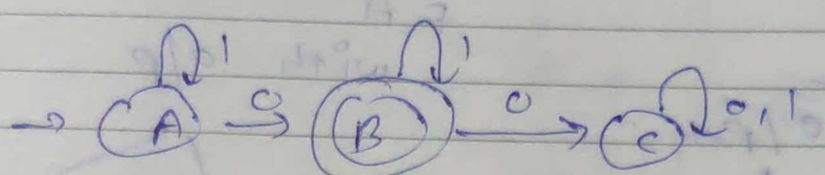
5) $L2 L3 = \{1110, 11011, 110, 1011\}$



Type – 0
Recursively enumerable grammar
Type – 1
Context sensitive grammar
Type – 2
Context free grammar
Type – 3
Regular grammar

Finite automata for language all string with only with contain exactly one 0

* Automata



A, B, C → states

○ → nonfinal state

◎ → final state (accepting state)

→○ → start state (or) initial state

- a machine M is for language L if we submit a string belong to L to machine M. Machine should accept by halting in one of the final state
- And if we submit a string which does not belong to Language L machine M should reject the string by halting in one of the non final state

1 1 0 1 1

$A \xrightarrow{1} A \xrightarrow{1} A \xrightarrow{0} B \xrightarrow{1} B \xrightarrow{1} B$

- FA
  - FA without o/p
    - DFA Deterministic finite Automata
    - NFA (NDFA) non-Deterministic finite Automata
      - NFA without ε-moves
      - NFA with ε move
  - FA with o/p
    - Moore Machine
    - Mealy Machine

* Deterministic FA :-
  is 5 tuple
  $M = (Q, \Sigma, \delta, q_0, F)$

  Q : is finite set state $\rightarrow \{A, B, C\}$

  Σ is input alphabet $\rightarrow \{0, 1\}$

  q_0 in Q starting state $\rightarrow \{A\}$

  $F \subseteq Q$ finite set of final state $\rightarrow \{B\}$

  δ is mapping from $Q \times \Sigma \rightarrow Q$

  $\{ABC\} \times \{0,1\}$